

# Numerical Simulation of an Untethered Omni-Directional Star-Shaped Swimming Robot

Xiaonan Huang<sup>1</sup>, Weicheng Huang<sup>2</sup>, Zachary Patterson<sup>1</sup>, Zhijian Ren<sup>1</sup>, M. Khalid Jawed<sup>2</sup> and Carmel Majidi<sup>1</sup>

**Abstract**—Simulating the swimming of soft underwater robot remains challenging due to the absence of an efficient numerical framework that can effectively capture the geometrically nonlinear deformation of soft materials and structures when interacting with a liquid environment. Here, we address this by introducing a discrete differential geometry-based model that incorporates an implicit treatment of the elasticity of soft limbs and a fluid model with three different components: hydrodynamic drag, jetting, and virtual added mass. The physical engine can run faster than real-time on a single thread desktop processor. We experimentally validate this numerical simulation tool by performing tests using an untethered omni-directional star-shaped swimming soft robot that is capable of moving with multiple swimming gaits. Quantitative agreement between experiment and simulation indicates the potential application of such a numerical framework for robot design and for model-based control schemes.

## I. INTRODUCTION

Soft robots represent an emerging class of biologically-inspired machines that are primarily composed of elastomers, fluids, and other forms of soft matter. They exhibit potential advantages over traditional piece-wise rigid machines in their ability to navigate through unstructured terrains [1], [2], squeeze through confined spaces [2], [3], and resist external disturbance [2], [4]. This is accomplished through their inherent elasticity and deformability. However, to realize the full potential of soft robots and achieve the ultimate goal of autonomous locomotion through challenging environments, it is necessary to develop a robust, accurate, and real-time computational framework. Such a simulation tool can provide guidelines for design and also be utilized in soft robot control and path planning. However, unlike the modeling of piecewise-rigid robots, which are composed of discrete components and limited degrees of freedom, the unlimited degrees of freedom in soft structures creates unique challenges in modeling the dynamics and locomotion of soft robots.

To achieve good agreement between reality and simulation of soft robot locomotion, considerable efforts have

been devoted to Finite Element Method [5]–[10], voxel-based simulation [11]–[13] and Cosserat rod theory-based modeling [14]–[16]. However, the computational cost of these approaches makes it challenging to use them for design optimization and real time closed-loop control for maneuvering tasks. Recently, a discrete differential geometry (DDG)-based method [17]–[19] for simulating soft robots has drawn considerable interest due to its improved computational efficiency for cases in which the robot has rod-like appendages [20]–[22]. The reliability of this DDG-based method in simulating on-land motion has been validated using a tethered fluidic driven star-fish inspired crawling robot [22] and a tethered shape memory alloy (SMA) driven star-shaped rolling robot [20]. It has also been used to optimize the geometric gait design of a tethered star-fish robot [22]. A key advantage of this DDG-based method is that the external force, contact, and collision models can be easily integrated into the discrete equations of motion. This provides opportunities to simulate underwater soft robots by adding a fluid-structure interaction model.

Here, we examine the ability to use DDG-based modeling to simulate dynamic locomotion of underwater soft robots. This is accomplished by adapting the numerical framework presented in Ref. [20], [23] to simulate an articulated, omni-directional, star-shaped swimming robot capable of multi-gait locomotion. The robot is designed with on-board miniaturized electronics and a battery in order to eliminate the need for external wiring, which could interfere with the hydrodynamics of swimming. Gravity is removed due to the buoyancy of the robot, and the fluid-structure interactions are accounted for by modeling forces that arise from hydrodynamic drag and jetting force, as developed in Ref. [24]. An “added mass” (interchangeably virtual mass) force from associated with non-hydrodynamic displacement

\*This work was supported by Army Research Office (Grant #: W911NF-16-1-0148; Program Manager: Dr. Samuel Stanton), the Office of Naval Research (ONR) under grant #: N00014-17-1-2063 (Program Manager: Dr. Tom McKenna) and the Henry Samueli School of Engineering and Applied Science, University of California, Los Angeles. (Xiaonan Huang, Weicheng Huang and Zach J. Patterson contribute equally and are co-first authors.) (Corresponding author: Carmel Majidi and M. Khalid Jawed.)

<sup>1</sup>Xiaonan Huang, Zach J. Patterson, Zhijian Ren and Carmel Majidi are with the Soft Machines Lab, Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA

<sup>2</sup>Weicheng Huang and M. Khalid Jawed are with the Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, 420 Westwood Plaza, Los Angeles, CA, 90095, USA

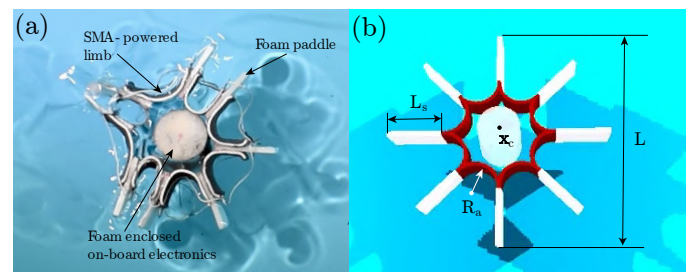


Fig. 1. (a) Components of the untethered star-shaped soft robot used as the experimental testbed. (b) Geometry of the untethered star-shaped soft robot as modeled in the computational simulation.

of surrounding fluid is also incorporated in the numerical framework. The actuation profile, defined by the change of curvature of a single actuator during one complete activation cycle, is fitted by a logistic function and then fed into the simulation in a manner similar to Ref. [20]. The resulting computational approach can run faster than real time – i.e. the computational run time on a single thread desktop processor is faster than the wall clock time of the physics that the engine is simulating. Overall, our results show reasonable quantitative agreement between the experiments and simulations and demonstrate that our discrete model represents a promising step toward simulation-based design and control of soft underwater robots.

## II. DESIGN OVERVIEW

In order to have an accurate comparison between the experiments and simulations, we first design and create an untethered, star-shaped, omni-directional soft robot. The robot is created by assembling eight radially connected SMA actuators (see Ref. [25] for the actuator design), as shown in Fig. 1(a). The tip of every actuator is connected to a rectangular piece of foam ( $35 \times 24 \times 3\text{mm}^3$ , 8722K9, McMaster) to ensure neutral buoyancy. Each actuator is controlled through a transistor (AO3416, Alpha & Omega Semiconductor Inc.) using an on-board microelectronic chip (Laird, BL652) that contains an nRF52832 microcontroller. The floating robot is powered by a 11.1V LiPo battery (300mAh, 45/75C, BETA FPV), which is encased along with the supporting printed circuit board within a foam (Soma Foama 15, Smooth-On) cylinder (height  $H_c = 50\text{mm}$ , radius  $R_c = 15\text{mm}$ , and 49g in weight); see Fig. 1(a) for a photo. The battery is selected to meet the requirement of simultaneously actuating at least three limbs. The Bluetooth enabled chip communicates via Bluetooth Low Energy (BLE) with an identical off-board chip. This off-board controller relays instructions transmitted from a computer through a UART connection. The computer generates motion patterns using a Python script that loops through a prescribed set of instructions. The diameter of the robot shown in Fig. 1(b) is  $L = 104\text{mm}$ , its weight is 91g, thickness is  $r_0 = 3.4\text{mm}$ . Here, we treat the in-plane height  $H$  as unit one due to the 2D simplification.

## III. NUMERICAL SIMULATION

In this section, we discuss the DDG-based numerical framework for the simulation of SMA-powered soft swimming robots. This framework is based on the method of Discrete Elastic Rods (DER) [18]. Starting from a geometric discretization of robotic structure, we formulate the equations of motion with a detailed discussion of both the internal forces derived from elastic strain energy and external forces from the interaction with the liquid environment. Then, we focus on modeling and electrical response of the SMA-based actuators used to power the robot limbs.

### A. Discrete model

Similar to the DDG-based method presented in [20], the numerical framework starts with a discrete representation of

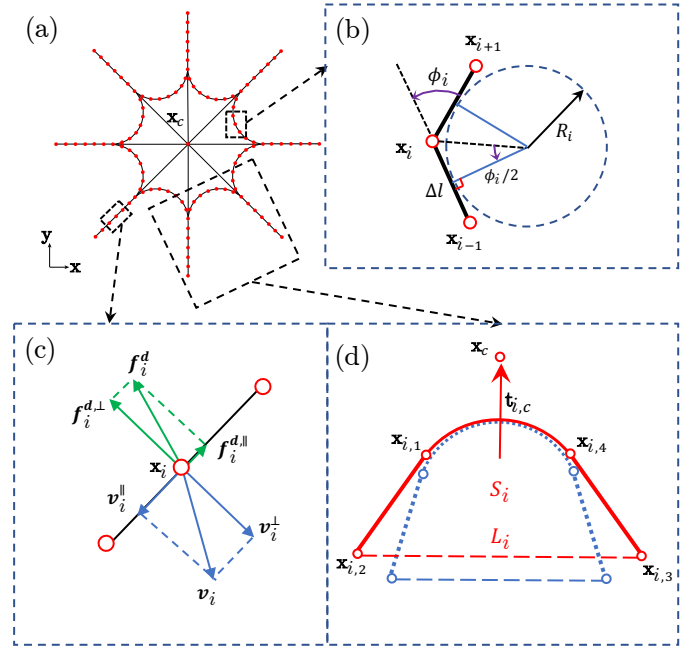


Fig. 2. (a) Discrete representation of the star-shaped robot. (b) Notations used in the discrete bending energy. (c) Formulation of drag force. (d) Schematic of jetting force generated by two adjacent foams.

the robot. The soft robot is modeled as a collection of elastic beams, shown schematically in Fig. 2(a). The whole structure is discretized into  $N$  nodes, and two neighboring nodes are connected with an edge. We model the soft robotic system as a mass-spring type system, with a lumped mass at each node, and associated elastic stretching and bending energies. Since the motion of the robot remains in 2D, a  $2N$  sized position vector  $\mathbf{q} = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_{N-1}]^T$  is used to describe the configuration of robot, where  $\mathbf{x}_i \equiv [x_i, y_i]^T$ .

Each rod segment between two consecutive nodes is treated as an edge that can stretch as the robot deforms – analogous to a linear spring. The relative rotation angle  $\phi_i$  (see Fig. 2(b)) between the two adjacent edges at each  $i$ -th node,  $\mathbf{x}_i$ , can change – similar to a torsional spring. The total elastic energy of the soft robot can be represented by the sum of two components: (1) stretching energies at each edge and (2) bending energy associated with each rotational angle. Note that the joint node is connected with four other nodes and, therefore, is related to four stretching energies and two bending energies.

The discrete stretching energy at the  $i$ -th edge connecting  $\mathbf{x}_i$  and  $\mathbf{x}_{i+1}$  is  $E_i^s = \frac{1}{2} E r_0 \varepsilon_i^2 \Delta l_i$ , where  $E r_0$  is the stretching stiffness in 2D and  $\varepsilon_i = |\mathbf{x}_{i+1} - \mathbf{x}_i| / \Delta l_i - 1$  is the uniaxial stretch strain. At the  $i$ -th rotational angle,  $\phi_i$ , the discrete bending energy is  $E_i^b = \frac{1}{2} \frac{1}{12} E r_0^3 (\kappa_i - \kappa_i^0)^2 \Delta l_i$ , where  $\kappa_i = 2 \tan(\phi_i/2) / \Delta l_i$  is the discrete curvature and  $\kappa_i^0$  is its natural curvature (i.e. the curvature evaluated in the undeformed configuration). The total elastic energy of the robot is  $E = \sum_i E_i^s + \sum_i E_i^b$ , such that the internal elastic force acting on  $i$ -th node,  $\mathbf{x}_i$ , can be obtained by computing the gradient of total potentials, i.e.  $\mathbf{f}_i^{\text{int}} = - \left[ \frac{\partial E}{\partial x_i}, \frac{\partial E}{\partial y_i} \right]^T$ . The

internal elastic force vector for the  $2N$  sized system is  $\mathbf{F}^{\text{int}} = [\mathbf{f}_0^{\text{int}}, \mathbf{f}_1^{\text{int}}, \dots, \mathbf{f}_{N-1}^{\text{int}}]^T$ .

Next, we turn to the formulation of the external forces. The external force is comprised of the force from the external environment, e.g., gravity or the damping force from a viscous medium. Here, we consider the hydrodynamic force from fluid experienced by soft robotic structure as the external forces in our numerical framework. The external forces can be formed by  $\mathbf{F}^{\text{ext}} = \mathbf{F}^d + \mathbf{F}^c + \mathbf{F}^j + \mathbf{F}^a$ , where  $\mathbf{F}^d$  is the drag force from fluid experienced by the soft limbs,  $\mathbf{F}^c$  is the drag force for the central cylinder,  $\mathbf{F}^j$  is the jetting force when the water between two neighboring limbs is squeezed out, and  $\mathbf{F}^a$  is the force associate with the added mass of the periodically accelerating surrounding fluid that displaces with the robot.

Shown schematically in Fig. 2(c), the drag force applied to the  $i$ -th node on soft limb, is given by [26], [27]

$$\mathbf{f}_i^d = -\frac{1}{2}\rho_f\Delta l_i \left( C_d^{\parallel}|\mathbf{v}_i^{\parallel}|\mathbf{v}_i^{\parallel} + C_d^{\perp}|\mathbf{v}_i^{\perp}|\mathbf{v}_i^{\perp} \right), \quad (1)$$

where  $\rho_f$  is the density of the fluid medium,  $C_d^{\parallel}$  and  $C_d^{\perp}$  are the drag coefficient along rod axial direction and normal direction, respectively,  $\mathbf{v}_i \equiv [\dot{x}_i, \dot{y}_i]^T$  is the velocity at  $i$ -th node, and  $\mathbf{v}_i^{\parallel}$  (and  $\mathbf{v}_i^{\perp}$ ) is its component along the segment tangent (and normal) direction. The drag force for the center cylinder can be formulated in a similar approach, i.e.,  $\mathbf{F}^c = -\frac{1}{2}\rho_f C_c D_c |\mathbf{v}_c| \mathbf{v}_c$ , where  $C_c$  is its drag coefficient and  $D_c = 2R_c$  is its diameter.

The jet force is associated with the change in volume between two neighboring rigid limbs (inter-limb space) as the limbs are actuated. Referring to Fig. 2(d), this change in volume (bounded by the solid red and dashed blue lines) causes the water to squeezed out, resulting in a propulsive jetting force. To simplify the modeling we apply this jetting force to the central node,  $\mathbf{x}_c$ , along each limb. The jet force for the  $i$ -th inter-limb space is [24], [28]

$$\mathbf{f}_i^j = -\frac{1}{L_i}\rho_f(\alpha\dot{S}_i)^2\mathbf{t}_{i,c}, \quad (2)$$

where  $L_i = |\mathbf{x}_{i,2} - \mathbf{x}_{i,3}|$  is the opening length of  $i$ -th inter-limb space,  $\mathbf{t}_{i,c}$  is the direction vector between the limb and the center,  $S_i$  is the area enclosed by  $\{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3}, \mathbf{x}_{i,4}\}$ ,  $\dot{S}_i$  is its changing rate, and  $\alpha$  is the ratio of the fluid jet out in plane.

Finally, we consider the added mass effect. The added mass is associated with the inertia of the surrounding fluid that moves with the robot. The added mass force at the  $i$ -th node is [28]

$$\mathbf{f}_i^a = -\rho_f C_a r_0 \Delta l_i \ddot{\mathbf{x}}_i, \quad (3)$$

where  $C_a$  is the added mass coefficient and  $\ddot{\mathbf{x}}_i \equiv [\ddot{x}_i, \ddot{y}_i]^T$  is its acceleration.

With both the internal forces and external forces formulated above, we now turn our focus to the time integration of star-shaped robotic system. At each time step,  $t_k$ , we know the DOF vector,  $\mathbf{q}(t_k)$ , and its velocity,  $\dot{\mathbf{q}}(t_k)$ . To obtain the positions and velocities at the next time step, we need to solve the force balance in order to determine  $\mathbf{q}(t_{k+1})$  and

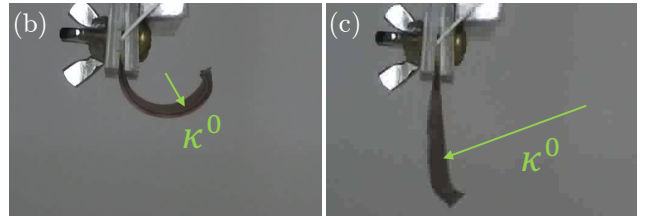
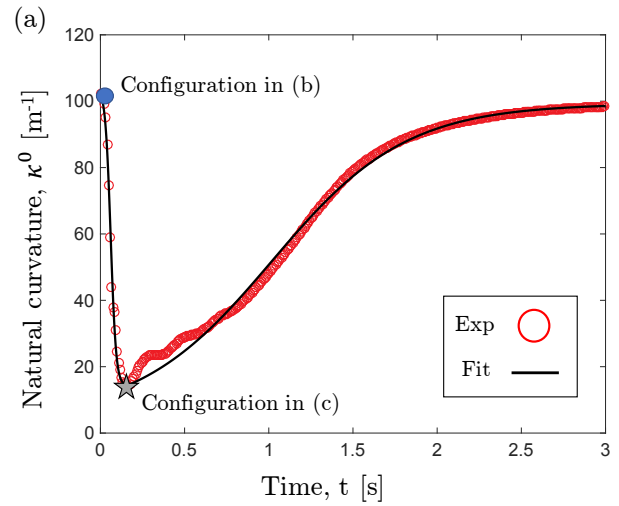


Fig. 3. (a) Experiment and fitting actuation profile of a single actuator. Configurations of a single actuator in (b) static phase and (c) actuated phase.

$\dot{\mathbf{q}}(t_{k+1})$ . A first order, implicit Euler integration is employed to update the equations of motion from  $t = t_k$  to  $t = t_{k+1}$ ,

$$\mathbb{M} \frac{\Delta \mathbf{q}(t_{k+1}) - h \dot{\mathbf{q}}(t_k)}{h^2} - \mathbf{F}^{\text{int}}(t_{k+1}) - \mathbf{F}^{\text{ext}}(t_{k+1}) = \mathbf{0} \quad (4a)$$

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \Delta \mathbf{q}(t_{k+1}) \quad (4b)$$

$$\dot{\mathbf{q}}(t_{k+1}) = \frac{\mathbf{q}(t_{k+1}) - \mathbf{q}(t_k)}{h}, \quad (4c)$$

where  $\mathbb{M}$  is the diagonal mass matrix,  $h$  is the time step size, and the  $t_{k+1}$  in brackets (or  $t_k$ ) indicates the quantities evaluated at  $t = t_{k+1}$  (or  $t = t_k$ ). The Newton-Raphson method is used to iteratively optimize the DOF vector until the error is within the tolerance.

### B. Model of Actuation

When a limb is actuated, its natural curvature significantly decreases in a short period of time; when it is deactivated, it slowly curls back to its original value. To include such actuation and deactivation in our model, we fit the curvature into a piece-wise function that has a logistic form (seeing Fig. 3(a) for details),

$$\bar{\kappa}(t) = \begin{cases} \frac{c_{1a}}{(1+e^{-c_{2a}(t-c_{3a})})} + c_{4a} & \text{when } t < t_0 \\ \frac{c_{1d}}{(1+e^{-c_{2d}(t-c_{3d})})} + c_{4d} & \text{when } t > t_0, \end{cases} \quad (5)$$

where  $c_{1a} - c_{4a}$  and  $c_{1d} - c_{4d}$  are fitted parameters for the actuation and deactuation profiles respectively. The parameters  $c_{1a} - c_{4a}$  and  $c_{1d} - c_{4d}$  are -89.827, 62.724, 0.061, 104.040 and 94.777, 2.507, 1.019, 4.445, respectively. The time interval  $t_0 = 0.163$  is the fitted actuation time (when curvature



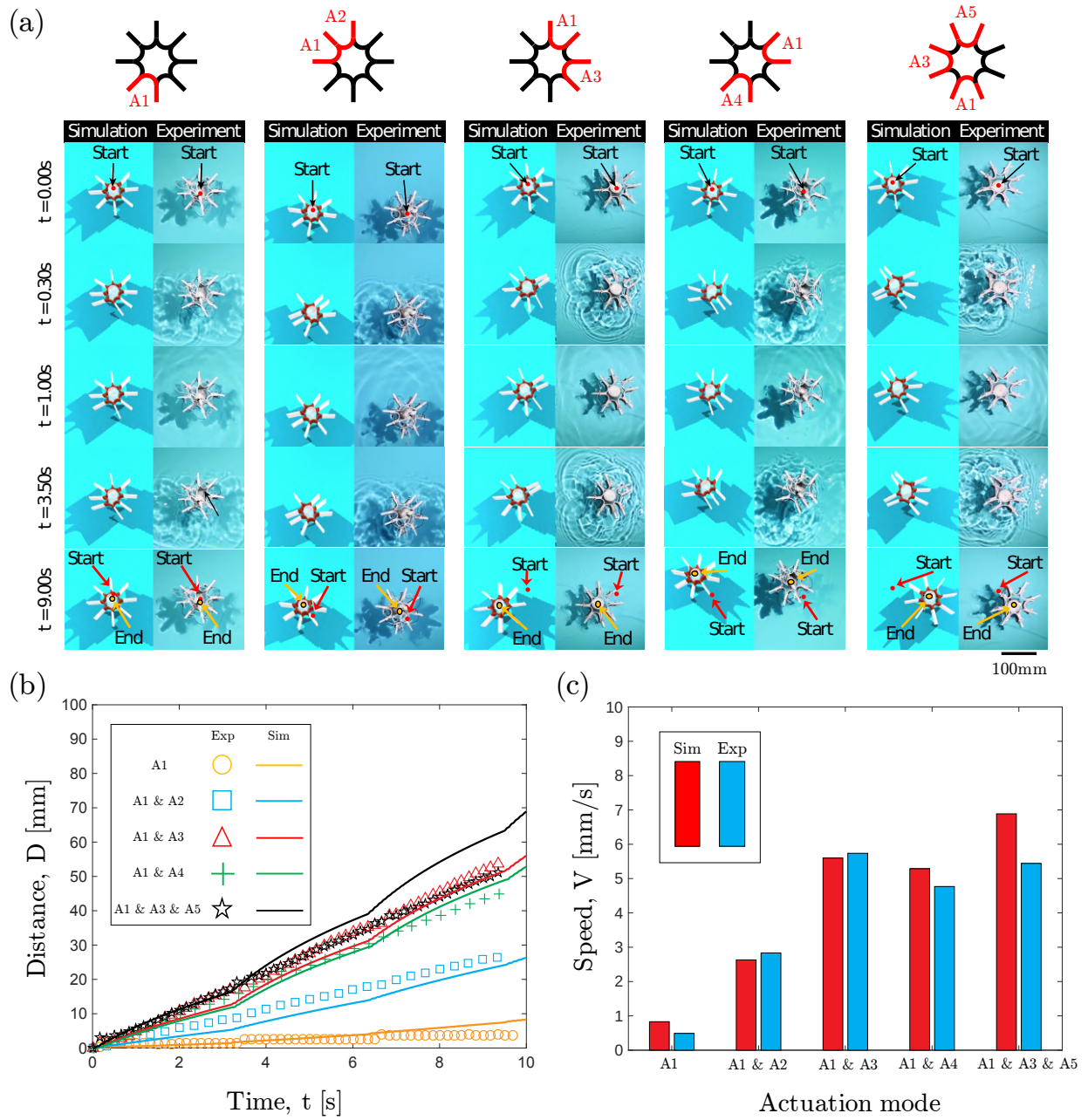


Fig. 4. (a) Snapshots of the star-shape robot in five different swimming gaits with actuation of (i) actuator A1, (ii) actuators A1 & A2, (iii) actuators A1 & A3, (iv) actuators A1 & A4, and (v) actuators A1 & A3 & A5. (b) The displacement of the robot centroid as a function of time for 3 cycles from both experiments (dashed lines) and simulations (solid lines). (c) The absolute speed comparison between simulations and experiments in five different swimming gaits

reaches the minimum value). The representative configurations of the initial and actuated profiles are presented in Fig. 3(b) and Fig. 3(c). The experimental characterization method is the same as the one presented in [20].

#### IV. RESULTS & DISCUSSION

Using the methods described in the previous sections, we are able to simulate various modes of robot locomotion and compare these predictions with experimental measurements. The computational modeling is based on the following

physical parameters, which are obtained from measurements of the experimental robot testbed. The effective density of the curved actuator and straight foam are  $2700\text{kg/m}^3$  and  $700\text{kg/m}^3$ , respectively. The density of water is  $1000\text{kg/m}^3$ . The Reynolds number in the current study is  $Re \approx 10^3$ , and the drag coefficients and added mass parameters we choose are  $C_d^{\parallel} = 0.01$ ,  $C_d^{\perp} = 1.0$ ,  $C_c = 10.0$ ,  $C_a = 1.0$  [28]. The coefficient  $\alpha$  is set as 0.8, which suggests that 80% of the enclosed fluid between adjacent limbs is jetted out through the aperture opening. The Young's modulus of the

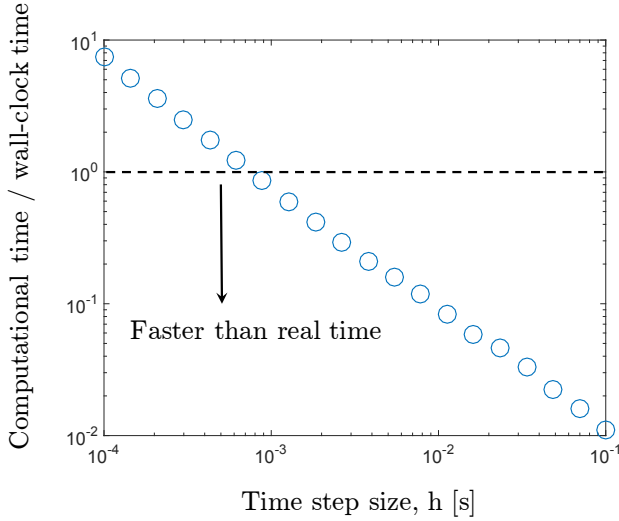


Fig. 5. The ratio between computational time and wall-clock time as a function of time step size,  $h$ .

soft actuator between each pair of protruding limbs is  $E = 5.0\text{MPa}$  (characterized by using the same method shown in Ref. [20]). The Young's modulus of the limbs themselves, on the other hand, is set to be  $100\times$  larger due to its natural rigidity. In the discrete setting, the nodal number is chosen to be  $N = 193$ , and the resulting discrete length of each segment is  $\Delta l_i \approx 1\text{mm}$  (except the edge connecting the center and the joint). The time step size used for the computational simulation is  $h = 1\text{ms}$ .

The locomotion of the robot is examined with five different swimming gait patterns. As shown in Fig. 4(a), the swimming directions are in agreement between the simulations and experiments for all five gaits. When A1 and A1 & A2 are actuated, the robot swims forward, suggesting that the propulsion provided by the drag force is larger than the thrust force. However, for cases when the thrust force dominates, the robot swims in the reverse direction. This happens for the following actuator combinations: A1 & A3, A1 & A4 and A1 & A3 & A5.

Figure 4(b) presents the displacement of the centroid of the robot with time over three actuation cycles for the five various actuator combinations. For the simplest scenario when a single limb is actuated, the drag force provides forward propulsion. Meanwhile, the water in the two adjacent actuators jets out, providing propulsion in the reverse direction. The two forces in this scenario are comparable so the robot moves at a very low speed ( $\sim 0.49\text{mm/s}$ ). Moreover, the simulation result agrees well with the experiment results in terms of speed. Next we consider the actuation of multiple limbs. When two adjacent limbs (A1 & A2) are actuated together, the joint edge does not move and the other two edges row to the sides, leading to the water in the neighbouring actuators jetting out. The direction of the combined drag force and thrust force are different from the first scenario and drag force dominates, resulting in a faster forward speed ( $\sim$

$2.83\text{mm/s}$ ). The simulation and experiment agree well with regard to swimming speed. When actuator A1 & A3, A1 & A4 and A1 & A3 & A5 are actuated, the water enclosed in at least three actuators jets out at a relatively high speed and generates a combined thrust force that dominates the robot motion. The robot swims at peak speed when A1 & A3 & A5 are actuated together. The simulation shows a similar trend but predicts higher speeds in the latter three scenarios compared with the experimental results. This overestimation could be attributed to the assumption that 80% of water jets out through the in-plane aperture opening between the limbs. In reality, more water is likely squeezed out vertically through the top and bottom openings. The exact value should be characterized experimentally. Another possible reason for the discrepancy is that the simulation is strictly planar while the robot undergoes out-of-plane pitch and roll due to the fluid turbulence and small mass distribution asymmetries.

Finally, we highlight the computational efficiency of our DER-based swimming robot simulator. In Fig. 5, with a fixed number of nodes,  $N = 193$ , the computational time linearly scales with time step size,  $h$ . This numerical framework can achieve real-time simulation speed – the time required to perform the computation is equal to or faster than the duration of the physical motion being modeled – when the time step size satisfies  $h \gtrsim 1.0\text{ms}$ . The simulations are performed on a single thread of Intel Core i7-6600U Processor @ 3.4 GHz.

## V. CONCLUSION

This work presents a computational framework adapted from the DER [20], [23] to simulate an underwater soft robot. The actuation profile of a single actuator is fed into the simulation and the combination of nonlinear elastic force, hydrodynamic drag force, jetting force and added mass force are considered in the numerical framework. To validate the simulation, we produce an untethered, omnidirectional, multi-limbed soft robot that is capable of moving with a variety of swimming gaits. The results show good agreement between the simulations and experiments in all five scenarios and thus validate the application of this DDG-based method in simulating multi-limbed underwater soft robots. This suggests that the numerical framework presented here can potentially be used in design optimization and simulation-based feedback control of multi-limbed swimming soft robots for maneuvering tasks.

## REFERENCES

- [1] M. T. Tolley, R. F. Shepherd, B. Mosadegh, K. C. Galloway, M. Wehner, M. Karpelson, R. J. Wood, and G. M. Whitesides, "A resilient, untethered soft robot," *Soft robotics*, vol. 1, no. 3, pp. 213–223, 2014.
- [2] X. Huang, K. Kumar, M. K. Jawed, A. M. Nasab, Z. Ye, W. Shan, and C. Majidi, "Chasing biomimetic locomotion speeds: Creating untethered soft robots with shape memory alloy actuators," *Science Robotics*, vol. 3, no. 25, p. eaau7557, 2018.
- [3] R. F. Shepherd, F. Ilievski, W. Choi, S. A. Morin, A. A. Stokes, A. D. Mazzeo, X. Chen, M. Wang, and G. M. Whitesides, "Multigait soft robot," *Proceedings of the national academy of sciences*, vol. 108, no. 51, pp. 20400–20403, 2011.

- [4] S. Seok, C. D. Onal, K.-J. Cho, R. J. Wood, D. Rus, and S. Kim, "Meshworm: a peristaltic soft robot with antagonistic nickel titanium coil actuators," *IEEE/ASME Transactions on mechatronics*, vol. 18, no. 5, pp. 1485–1497, 2012.
- [5] Y. Fei and H. Xu, "Modeling and motion control of a soft robot," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1737–1742, 2016.
- [6] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.
- [7] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *2013 IEEE international conference on robotics and automation*, pp. 3982–3987, IEEE, 2013.
- [8] G. Runge and A. Raatz, "A framework for the automated design and modelling of soft robotic systems," *CIRP Annals*, vol. 66, no. 1, pp. 9–12, 2017.
- [9] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1576, 2018.
- [10] J. Chenevier, D. González, J. V. Aguado, F. Chinesta, and E. Cueto, "Reduced-order modeling of soft robots," *PloS one*, vol. 13, no. 2, p. e0192052, 2018.
- [11] J. Hiller and H. Lipson, "Dynamic simulation of soft multimaterial 3d-printed objects," *Soft robotics*, vol. 1, no. 1, pp. 88–101, 2014.
- [12] N. Cheney, J. Bongard, and H. Lipson, "Evolving soft robots in tight spaces," in *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*, pp. 935–942, 2015.
- [13] S. Kriegman, A. M. Nasab, D. Shah, H. Steele, G. Branin, M. Levin, J. Bongard, and R. Kramer-Bottiglio, "Scalable sim-to-real transfer of soft robot designs," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 359–366, IEEE, 2020.
- [14] X. Zhou, C. Majidi, and O. M. O'Reilly, "Soft hands: An analysis of some gripping mechanisms in soft robot design," *International Journal of Solids and Structures*, vol. 64, pp. 155–165, 2015.
- [15] S. Grazioso, G. Di Gironimo, and B. Siciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *Soft robotics*, vol. 6, no. 6, pp. 790–811, 2019.
- [16] F. Renda, F. Giorgio-Serchi, F. Boyer, C. Laschi, J. Dias, and L. Seneviratne, "A unified multi-soft-body dynamic model for underwater soft robots," *The International Journal of Robotics Research*, vol. 37, no. 6, pp. 648–666, 2018.
- [17] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, "Discrete elastic rods," in *ACM SIGGRAPH 2008 papers*, pp. 1–12, 2008.
- [18] M. Bergou, B. Audoly, E. Vouga, M. Wardetzky, and E. Grinspun, "Discrete viscous threads," in *ACM Transactions on Graphics (TOG)*, vol. 29, p. 116, ACM, 2010.
- [19] M. K. Jawed, A. Novelia, and O. M. O'Reilly, *A primer on the kinematics of discrete elastic rods*. Springer, 2018.
- [20] W. Huang, X. Huang, C. Majidi, and M. K. Jawed, "Dynamic simulation of articulated soft robots," *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020.
- [21] N. N. Goldberg, X. Huang, C. Majidi, A. Novelia, O. M. O'Reilly, D. A. Paley, and W. L. Scott, "On planar discrete elastic rod models for the locomotion of soft robots," *Soft robotics*, vol. 6, no. 5, pp. 595–610, 2019.
- [22] W. L. Scott and D. A. Paley, "Geometric gait design for a starfish-inspired robot using a planar discrete elastic rod model," *Advanced Intelligent Systems*, p. 1900186, 2020.
- [23] W. Huang, Z. Patterson, C. Majidi, and M. K. Jawed, "Modeling soft swimming robots using discrete elastic rod method," in *Bioinspired Sensing, Actuation, and Control in Underwater Soft Robotic Systems*, pp. 247–259, Springer, 2021.
- [24] S. P. Colin and J. H. Costello, "Morphology, swimming performance and propulsive mode of six co-occurring hydromedusae," *Journal of experimental biology*, vol. 205, no. 3, pp. 427–437, 2002.
- [25] X. Huang, K. Kumar, M. K. Jawed, A. Mohammadi Nasab, Z. Ye, W. Shan, and C. Majidi, "Highly dynamic shape memory alloy actuator for fast moving soft robots," *Advanced Materials Technologies*, vol. 4, no. 4, p. 1800540, 2019.
- [26] Y. Modarres-Sadeghi, M. Païdoussis, and C. Semler, "A nonlinear model for an extensible slender flexible cylinder subjected to axial

- flow,” *Journal of Fluids and Structures*, vol. 21, no. 5-7, pp. 609–627, 2005.
- [27] M. Mallick, A. Kumar, N. Tamboli, A. Kulkarni, P. Sati, V. Devi, and S. Chandar, “Study on drag coefficient for the flow past a cylinder,” *International Journal of Civil Engineering Research*, vol. 5, no. 4, pp. 301–306, 2014.
- [28] E. Kelasidi, K. Y. Pettersen, J. T. Gravdahl, and P. Liljebäck, “Modeling of underwater snake robots,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4540–4547, IEEE, 2014.